Junaid Girkar

60004190057

TE COMPS A4

EXPERIMENT - 7

**Aim: Make use of RE module to perform text processing**

Theory:

A Regular Expressions (RegEx) is a special sequence of characters that uses a search pattern to find a string or set of strings. It can detect the presence or absence of a text by matching with a particular pattern, and also can split a pattern into one or more sub-patterns. Python provides a re module that supports the use of regex in Python. Its primary function is to offer a search, where it takes a regular expression and a string. Here, it either returns the first match or else none.

## RegEx Functions

The re module offers a set of functions that allows us to search a string for a match:

| **Function** | **Description** |
| --- | --- |
| findall | Returns a list containing all matches |
| search | Returns a Match object if there is a match anywhere in the string |
| split | Returns a list where the string has been split at each match |
| sub | Replaces one or many matches with a string |

## Metacharacters

Metacharacters are characters with a special meaning:

| **Character** | **Description** | **Example** |
| --- | --- | --- |
| [] | A set of characters | "[a-m]" |
| \ | Signals a special sequence (can also be used to escape special characters) | "\d" |
| . | Any character (except newline character) | "he..o" |
| ^ | Starts with | "^hello" |
| $ | Ends with | "planet$" |
| \* | Zero or more occurrences | "he.\*o" |
| + | One or more occurrences | "he.+o" |
| ? | Zero or one occurrences | "he.?o" |
| {} | Exactly the specified number of occurrences | "he{2}o" |
| | | Either or | "falls|stays" |
| () | Capture and group |  |

## Special Sequences

A special sequence is a \ followed by one of the characters in the list below, and has a special meaning:

| **Character** | **Description** | **Example** |
| --- | --- | --- |
| \A | Returns a match if the specified characters are at the beginning of the string | "\AThe" |
| \b | Returns a match where the specified characters are at the beginning or at the end of a word  (the "r" in the beginning is making sure that the string is being treated as a "raw string") | r"\bain"  r"ain\b" |
| \B | Returns a match where the specified characters are present, but NOT at the beginning (or at the end) of a word  (the "r" in the beginning is making sure that the string is being treated as a "raw string") | r"\Bain"  r"ain\B" |
| \d | Returns a match where the string contains digits (numbers from 0-9) | "\d" |
| \D | Returns a match where the string DOES NOT contain digits | "\D" |
| \s | Returns a match where the string contains a white space character | "\s" |
| \S | Returns a match where the string DOES NOT contain a white space character | "\S" |
| \w | Returns a match where the string contains any word characters (characters from a to Z, digits from 0-9, and the underscore \_ character) | "\w" |
| \W | Returns a match where the string DOES NOT contain any word characters | "\W" |
| \Z | Returns a match if the specified characters are at the end of the string | "Spain\Z" |

## Sets

A set is a set of characters inside a pair of square brackets [] with a special meaning:

| **Set** | **Description** |
| --- | --- |
| [arn] | Returns a match where one of the specified characters (a, r, or n) are present |
| [a-n] | Returns a match for any lower case character, alphabetically between a and n |
| [^arn] | Returns a match for any character EXCEPT a, r, and n |
| [0123] | Returns a match where any of the specified digits (0, 1, 2, or 3) are present |
| [0-9] | Returns a match for any digit between 0 and 9 |
| [0-5][0-9] | Returns a match for any two-digit numbers from 00 and 59 |
| [a-zA-Z] | Returns a match for any character alphabetically between a and z, lower case OR upper case |
| [+] | In sets, +, \*, ., |, (), $,{} has no special meaning, so [+] means: return a match for any + character in the string |

Code:

| import re text\_to\_search = '''abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890 123.456.789 123-456-789 123\*456\*789 cat  mat bat  Mr. Smith Mr David Mrs. Riya Mr. Ha HaHa '''   def pattern\_finder(pattern, texts):  for text in texts.split('\n'):  matches = pattern.finditer(text)  for match in matches:  print(match.group())   pattern\_finder(re.compile(r'abc'), text\_to\_search) pattern\_finder( re.compile(r'^[a-zA-Z]'), text\_to\_search) pattern\_finder(re.compile(r'[^b]at'), text\_to\_search) pattern\_finder(re.compile(r'\d\d\d'), text\_to\_search) pattern\_finder(re.compile(r'\d{3}.\d{3}.\d{3}'), text\_to\_search) pattern\_finder(re.compile(r'Mr\.?'), text\_to\_search) pattern\_finder(re.compile(r'Mr\.?\s[A-Z]\w\*'), text\_to\_search) pattern\_finder(re.compile(r'M(r|s|rs).?\s[A-Z]\w\*'), text\_to\_search)   # #EMAILS emails = junaid@gmail.com junaid.girkar@gmail.com junaid-123-girkar@gmail.com ''' pattern\_finder(re.compile(r'[a-zA-Z0-9.-].com'), emails) pattern\_finder(re.compile(r'[a-zA-Z0-9.-]+@[a-zA-Z-]+\.com'), emails) pattern\_finder(re.compile(r'[a-zA-Z0-9.-]+@[a-zA-Z-]+\.(com|ac|net)'), emails)  # URLS urls = ''' https://www.google.com https://youtube.com http://djsce.ac.in https://www.nasa.gov ''' pattern\_finder(re.compile(r'http'), urls) pattern\_finder(re.compile(r'https?'), urls) pattern\_finder(re.compile(r'https?://(www\.)?'), urls)  # Use of Group (index) pattern=re.compile(r'https?://(www\.)?(\w+)(\.\w+)') matches = pattern.finditer(urls) for match in matches:  print(match.group(0))  print(match.group(2))  print(match.group(3))  # Use of Sub subbed\_urls=pattern.sub(r'\1\2\3',urls) print(subbed\_urls) |
| --- |

Output:

| abc a A c m b M M M M cat mat 123 456 789 123 456 789 123 456 789 123 456 789 123.456.789 123-456-789 123\*456\*789 Mr. Mr Mr Mr. Mr. Smith Mr David Mr. Ha Mr. Smith Mr David Mrs. Riya Mr. Ha l.com o.com junaid@gmail.com junaid-123-girkar@gmail.com junaid@gmail.com junaid.girkar@gmail.com junaid-123-girkar@gmail.com http http http http https https http https https://www. https:// http:// https://www. https://www.google.com google .com https://youtube.com youtube .com http://djsce.ac djsce .ac https://www.nasa.gov nasa .gov  www.google.com youtube.com djsce.ac.in www.nasa.gov |
| --- |

**Conclusion:**

We learnt about regular expressions and its workings. We then looked at its applications and used it in our python program.